In the Specification

Please amend the specification of this application as follows:

Rewrite the paragraph at page 2, lines 11 to 22 as follows:

--The present invention deals with the write request queue as applied to the transfer controller with hub and ports architecture. The transfer controller with hub and ports is the subject of U.K. Patent Application serial number ~~_____~~, 9909195.9 filed April 16, 1999 (TI-28983UK) having a U. S. counterpart application, now U.S. Patent No. 6,496,740. The transfer controller with hub and ports is a significant basic improvement in data transfer techniques in complex digital systems and provides, along with many other features, for the uniform implementation of port interfaces at the periphery of such systems. Some of these ports may be slow, i.e. they may have relatively low throughput. Others may be relatively fast, having the throughput and speed of a current central processing unit.--

Rewrite the paragraph at page 7, line 18 to page 8, line 2 as follows:

--Figure 1 illustrates a block diagram of the basic features of the transfer controller with hub and ports. The transfer controller with hub and ports is basically a data transfer controller which has at its front end portion, a queue manager 100 receiving, prioritizing, and dispatching data in the form of transfer request packets. This queue manager 100 connects within the hub unit 110 to the channel registers 120. Channel registers ~~12~~ 120 receives the data transfer request packets and processes them first by prioritizing them and assigning them to one of the N channels. Each channel represents a priority level. These channel registers 120 interface with the source control pipeline 130 and destination

control pipeline 140. These are address calculation units for source (read) and destination (write) operations.

Rewrite the paragraph at page 11, line 6 to page 12, line 2 as follows:

--Consider the following prior art example illustrated in Figure 3. There are two transfers operating concurrently, one between ports A 300 and B 301, and another between ports A 300 and C 302. The transfer between port A 300 and port B 301 has a higher priority 304 than the transfer between port a 300 and port C 302. The interface at port A 300 has a higher bandwidth 305 than both the bandwidth 306 of port B 310 and the bandwidth 307 of port C 302. When the transfer from port A 300 to port B 301 starts up, data read from port A 300 backlogs within the device because port B 301 is slower than port A 300. New requests to port A 300 can only be submitted at the rate that port B 301 is clearing the backlog. The interface at port A 300 cannot be used for transfers between port A 300 and port C 302 during the times that the interface at port A 300 cannot be used for fetching data for port B 301. This is a blocking transfer. When transferring data from port A 300 to port B 301, the prior art approach is to read from port A 300, and when data is returned, pass it on to port B 301. However, if the interface at port A 300 has a higher bandwidth than that of port B 301, this can result in a large backlog of data requested from port A 300 which cannot yet be passed on to port B 301. If transfer controller is multitasking, this backlog can prevent the interface at port A 300 from being applied to another transfer, such as the lower priority from port A 300 to port C 302 concurrently as illustrated in Figure 3.--

Rewrite the paragraph at page 18, lines 12 to page 19, line 19 as follows:

--Figure 8 illustrates from a higher level an overview of ~~an~~ a multiprocessor integrated circuit employing the transfer controller with hub and ports of this invention. There are four main functional blocks. The transfer controller with hub and ports 110 and the ports including ports external port interface units 240 to 247 and internal memory port 250 are the first two main functional blocks. The other two main functional blocks are the transfer request feed mechanism 201 and the data transfer bus (DTB) 255. These are closely associated functional units that are ~~but~~ not a part of the transfer controller with hub and ports 110. Transfer request feed mechanism 201 is coupled to plural internal memory port nodes 870, 871 and 872. Each of these internal memory port nodes includes an independently programmable data processor, which may be a digital signal processor, and corresponding cache memory or other local memory. The internal construction of these internal memory port nodes is not important for this invention. For the purpose of this invention it is sufficient that each of the internal memory port nodes can submit transfer requests via transfer request feed mechanism 201 and has memory that can be a source or destination for data. Transfer request feed mechanism 201 prioritizes these packet transfer requests in a manner not relevant to this invention. Transfers originating from or destined for internal memory port nodes 870, 871 or 872 are coupled to transfer controller with hub and ports 110 via data transfer bus 255 and internal memory port master 250. As previously described, internal memory port master 250 may not require the write driven process of this invention if internal memory port nodes 870, 871 and 872 have memory transfer bandwidth equivalent to the memory transfer bandwidth of transfer controller with hub and ports 110. Figure 8 highlights the possible connection of data transfer bus 255 to multiple internal memory port nodes 870, 871 and 872 and the possible connection of multiple transfer request nodes to transfer

request feed mechanism 201. This represents an example of the mode of use of the write driven process of this invention and not its only context of use.--